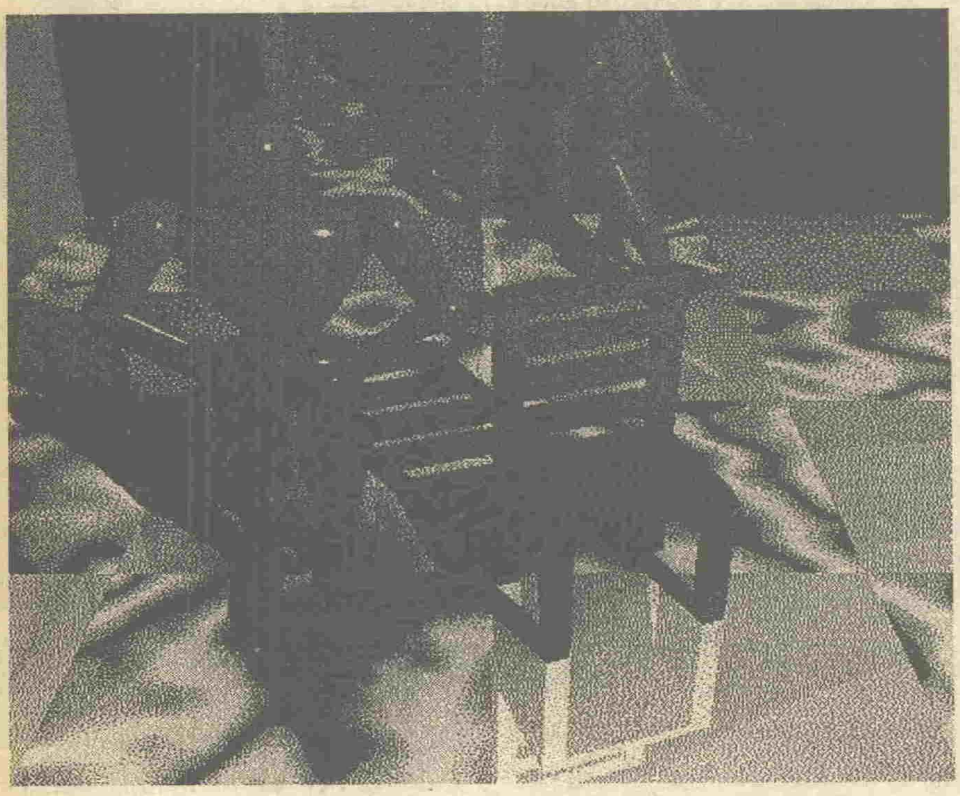
เทคโนโลยีคอมพิวเตอร์ กราฟิก : ทดลองใช้ Raytracer



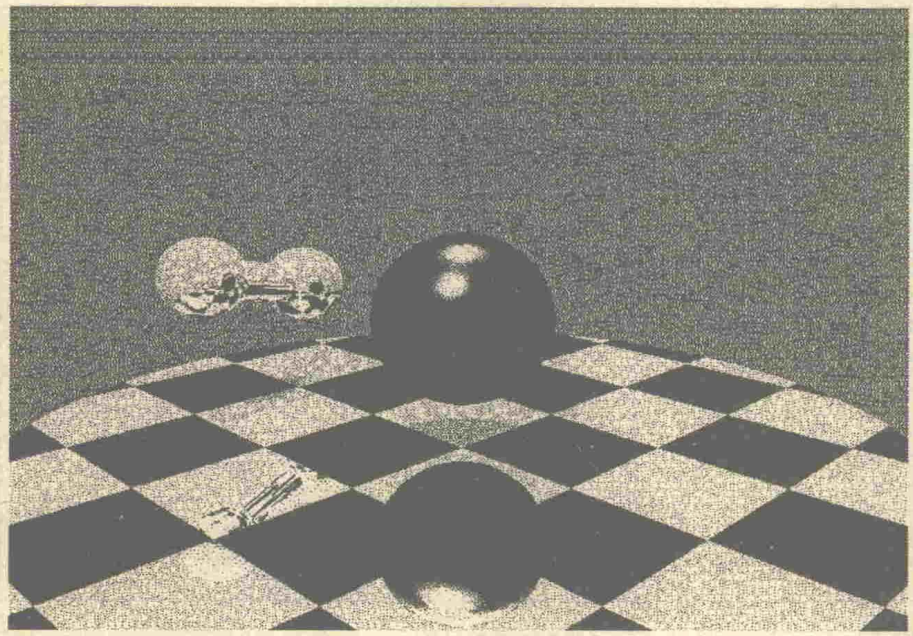
ทบทวนความจำคร่าว ๆ ในหลักการตามรอยเส้นแสง ตามด้วยแนวคิดในการสร้างภาพจากโปรแกรม Raytracer

Raytracing เป็นวิธีการสร้างภาพที่เหมือนจริงของวัตถุในโลกสามมิติบนจอภาพคอมพิวเตอร์โดยใช้หลักการตามรอยเส้นแสง ผู้อ่านบางท่านคงจะเคยได้เห็นภาพที่สร้างจากวิธีการนี้แล้ว ในช่วงแรก ๆ ที่มีภาพเหล่านี้ออกมาก็เป็นที่ฮือฮากันพอสมควร อาจจะเป็นเพราะว่าเป็นภาพที่ดูแล้วแปลกตา จะว่าเหมือนจริงก็ไม่ทีเดียวนัก บางภาพดูเหมือนเป็นภาพจากจินตนาการจากในนิยาย รูปที่ 1 เป็นตัวอย่างภาพที่สร้างจากวิธีนี้

หลักการสร้างภาพด้วยวิธีนี้ ผมได้กล่าวไว้บ้างแล้วในบทความชุดเดียวกันนี้ในหนังสือไมโครฯ ฉบับที่ 105 ในฉบับนี้ผมจะแนะนำโปรแกรม Raytracer โปรแกรมหนึ่งให้ท่านผู้อ่านทดลองนำไปสร้างภาพเล่นกัน เนื้อหาของบทความจะเริ่มจากการทบทวนความจำอย่างคร่าว ๆ ในเรื่องของหลักการของ Raytracing ตามด้วยแนวความคิดที่จำเป็นต้องใช้ในการสร้างภาพด้วยโปรแกรม Raytracer สุดท้ายจะเป็นการแนะนำและตัวอย่างการใช้งานโปรแกรม Raytracer ที่มีชื่อว่า Polyray

หลักการของการตามรอยเส้นแสง

หลักการของการสร้างภาพโดยใช้การตามรอยเส้นแสงนั้นใช้หลักการเลียนแบบการมองเห็นภาพที่ตาของมนุษย์นี่เอง ตาของคนเราสามารถมองเห็นสิ่งต่าง ๆ ได้นั้นเกิดจากการที่มีรังสีของแสงซึ่งส่งออกมาจากแหล่งกำเนิดแสง ส่องลงมากระทบกับวัตถุนั้น ๆ และสะท้อนมาเข้าตาของเรา รูปทรงและสีสันที่เรามองเห็นนั้นเกิดจากการที่แสงเปลี่ยนคุณสมบัติไปตามธรรมชาติแล้ว แสงมีคุณสมบัติมากมาย แสงจากดวงอาทิตย์ที่เราเห็นกันทุกวันนั้นประกอบด้วยเส้นรังสีเป็นจำนวนมาก เส้นรังสีแต่ละเส้นก็เดินทางเป็นเส้นตรง (ยกเว้นว่าจะเดินทางไปชนอะไรเข้า) ในแสงสีขาวก็ประกอบไปด้วยแสงสีต่างกันเจ็ดสี เราจะมองเห็นสีทั้งเจ็ดได้ก็ต่อเมื่อแสงเกิดการหักเห เช่น ตอนที่เกิดรุ้งกินน้ำ การหักเหและการสะท้อนก็เป็นคุณสมบัติอีกอย่างหนึ่งของแสง คุณสมบัติเหล่านี้ก็มีการค้นพบและสรุปออกมาเป็นหลักเกณฑ์ที่แน่นอน สามารถ



รูปที่ 1 ภาพที่สร้างจากโปรแกรม Polyray

นำมาใช้อธิบายปรากฏการณ์ต่าง ๆ ในเรื่องของแสงได้ เช่น ถ้าแสงสีขาวส่องกระทบวัตถุสีแดง แปลว่าวัตถุนั้นจะมีความสามารถในการดูดกลืนสีอื่น ๆ ยกเว้นสีแดง ดังนั้นแสงที่ตกกระทบวัตถุจะมีคุณสมบัติเปลี่ยนไปต่างไปจากแสงที่สะท้อนออกมา คุณสมบัติของแสงจะเปลี่ยนไปอย่างไรขึ้นกับคุณสมบัติของวัตถุตรงจุดที่แสงกระทบและคุณสมบัติของตัวแสงเอง นี้เป็นหลักการของการมองเห็น

โปรแกรม Raytracer ก็ใช้หลักการเดียวกัน โปรแกรม Raytracer สามารถเลียนแบบการเดินทางของแสงได้ และเมื่อเราจับเอาคุณสมบัติของแสงกับคุณสมบัติของวัตถุมาเขียนเป็นโปรแกรม เราก็จะได้โปรแกรมที่สามารถสร้างภาพที่เหมือนจริงขึ้นมา

ในรายละเอียดแล้ว สิ่งที่โปรแกรม Raytracer แตกต่างไปจากการสร้างภาพโดยธรรมชาติมีอยู่หลายประการคือ

- ในความเป็นจริงเมื่อเราเปิดสวิตช์ไฟจะมีรังสีของแสงเดินทางออกจากแหล่งกำเนิดทุกทิศทุกทางในเวลาเดียวกันด้วยความเร็วที่สูงมาก ดังนั้นเมื่อเราเปิดไฟ เราจะเห็นสิ่งต่าง ๆ รอบตัวได้ทันที แต่เนื่องจากในเครื่อง PC ธรรมดามีหน่วยประมวลผลอยู่เพียงตัวเดียว ดังนั้นโปรแกรม Raytracer ต้องทำการสร้างเส้นรังสีของแสงขึ้นมาทีละเส้นและทำการคำนวณตลอดเวลาว่าเส้นรังสีนั้นได้วิ่งไปชนวัตถุอะไรแล้วบ้าง ชนแล้วมีคุณสมบัติเปลี่ยนไปอย่างไร การที่ต้องทำการคำนวณเส้นรังสีทีละเส้นและความซับซ้อนของการคำนวณ ทำให้การทำงานของโปรแกรมช้ามาก

- ในตามธรรมชาติแสงจะเดินทางจากแหล่งกำเนิดมาเข้าตาของเรา แสงที่ออกจากแหล่งกำเนิดจะมีเป็นจำนวนมาก เช่น แสงจากหลอดไฟแบบกลมจะส่งออกไปได้รอบตัว ส่วนที่จะสะท้อนเข้าตาเราเป็นเพียงส่วนหนึ่งเท่านั้น ในการสร้างภาพด้วยคอมพิวเตอร์นั้นเราไม่สามารถที่จะทำการคำนวณการเดินทางของเส้นรังสีทั้งหมดได้ เพราะจะใช้เวลานานมาก เราจึงต้องใช้วิธีคำนวณเฉพาะส่วนที่จำเป็นเท่านั้น คือใช้การส่งเส้นรังสีออกจากตาผ่านจอภาพไปชนวัตถุแทน ความแตกต่างนี้ไม่ทำให้เกิดความแตกต่างขึ้นในภาพเลย เพราะการที่โปรแกรมใช้วิธีส่งเส้นรังสีย้อนทิศทางจากความเป็นจริงในธรรมชาตินั้นก็เพื่อลดจำนวนเส้นรังสีซึ่งจะมีผลทำให้เวลาการคำนวณลดลงเท่านั้น ส่วนการคำนวณการเปลี่ยนคุณสมบัติของแสงนั้นยังคงต้องใช้การคำนวณในทิศทางเดิมคือ ต้องเริ่มจากทางด้านของวัตถุไม่ใช่เริ่มจากตา เช่น ถ้าเส้นที่ผ่านจุดสี (pixel) ของจอภาพที่ตำแหน่ง A ไม่ได้ชนวัตถุใดและไม่ได้ชนแหล่งกำเนิดแสงเลย แสดงว่าไม่มีเส้นแสงจากแหล่งกำเนิดแสงใด ๆ ผ่านเข้ามายังตาเช่นกัน ดังนั้นจุดสี A ก็ควรจะเป็นสีดำ

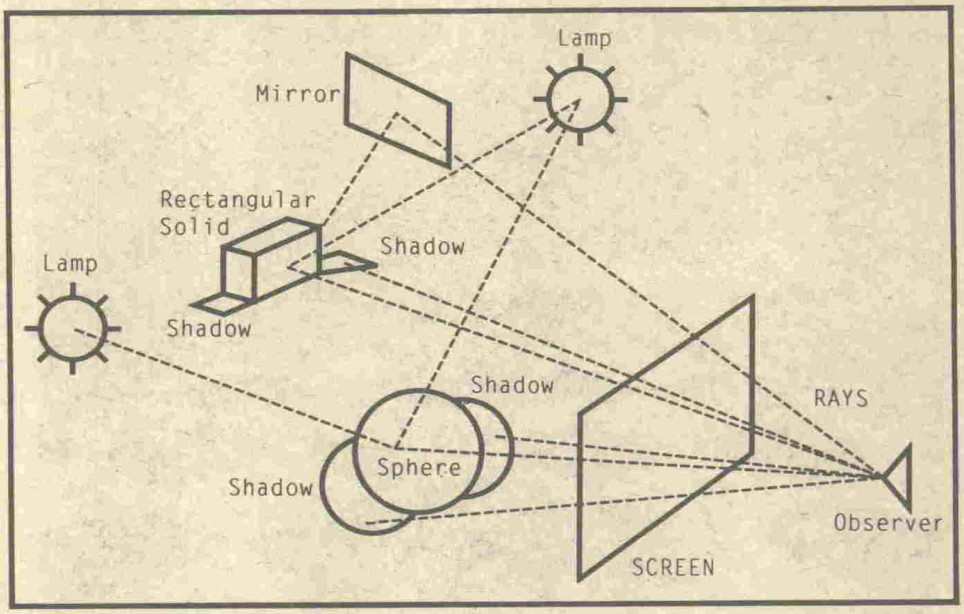
- ในธรรมชาติเราจะถือว่าเส้นรังสีของแสงนั้นมีแต่ความยาวเท่านั้น ไม่มีความหนาและความสูง มีเพียงมิติเดียว ทำให้เส้นรังสีของแสงที่ส่องไปยังพื้นที่เล็ก ๆ มีจำนวนเป็นอนันต์ แต่เนื่องจากเครื่องคอมพิวเตอร์สามารถแสดงรายละเอียดได้เล็กมากที่สุดก็ในระดับของจุดสี (Pixel) บนจอภาพเท่านั้น ดังนั้นโปรแกรม Raytracer จะทำการคำนวณโดยส่งเส้นรังสีจากตาทะลุผ่านจุดสีแต่ละจุดบนจอภาพออกไปยังวัตถุ จำนวนจุดสีบนจอภาพมีเท่าใดจำนวนเส้นรังสีก็มีเท่านั้น ดังแสดงในรูปที่ 2 ผลของความแตกต่างก็คือความคมชัดของภาพจะลดลง และจะมีผลมากในกรณีที่ความละเอียดของจอภาพต่ำ (ขนาดของจุดสีใหญ่) และวัตถุในภาพมีขนาดเล็ก โปรแกรม Raytracer บางตัวจะสามารถกำหนดจำนวนของเส้นรังสีที่โปรแกรมจะสร้างต่อจุดสีได้

- เนื่องจากต้องสร้างภาพบนจอภาพของเครื่องคอมพิวเตอร์ ดังนั้นใน

รูปแบบหรือโมเดลในการสร้างภาพโดยวิธีนี้จะมีจอภาพซึ่งทำหน้าที่เสมือนฉากรับภาพเข้ามาเกี่ยวข้องด้วย

นอกจากความแตกต่างที่กล่าวแล้วข้างต้น ยังมีรายละเอียดปลีกย่อยอื่น ๆ อีก ขึ้นกับว่าโปรแกรมจะทำการคำนวณคุณสมบัติของแสงได้ครบถ้วนเท่าใด เช่น ถ้าวัตถุในภาพเป็นปริซึม จะสามารถคำนวณการเกิดรุ้งได้หรือไม่

รูปที่ 2 จะแสดงส่วนประกอบโดยละเอียดของโมเดลในการสร้างภาพด้วยวิธี Raytracing ในภาพ Screen คือจอภาพของเครื่องคอมพิวเตอร์ Observer คือผู้สังเกต RAYS คือเส้นรังสีของแสง Lamp คือแหล่งกำเนิดแสง เส้นประคือเส้นรังสีของแสง กล่องสี่เหลี่ยม (Rectangular Solid) กระจกเงา (Mirror) และทรงกลม (Sphere) คือวัตถุที่อยู่ในภาพ วัตถุเหล่านี้มีความแตกต่างกันทั้งในด้านรูปทรงและพื้นผิว กระจกเงาจะสามารถสะท้อนแสงที่มาตกกระทบได้ดีมาก ดังนั้นเมื่อมีแสงที่เดินทางมาจากวัตถุอื่นมากระทบและสะท้อนมายังผู้สังเกต ผู้สังเกตจะมองเห็นวัตถุนั้นด้วย ถ้าวัตถุนั้นทึบแสงก็จะเกิดบริเวณของเงา (Shadow) ขึ้น แสงจากบริเวณเงาจะให้กำเนิดเส้นแสงที่มีความเข้มน้อยมากหรือไม่มีเลย ดังนั้นจุดสี ณ ตำแหน่งที่เส้นนั้นผ่านก็จะเป็นสีเข้มหรือสีดำ

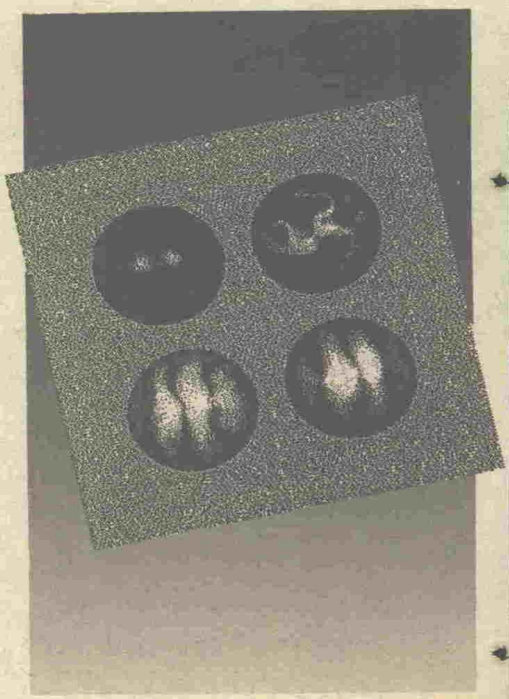


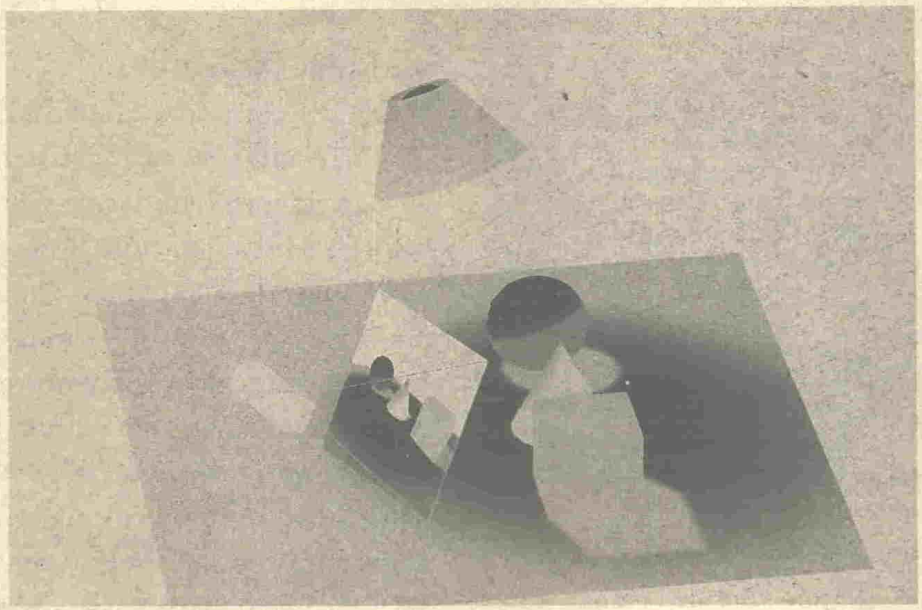
รูปที่ 2 หลักการสร้างภาพโดยวิธีการตามรอยเส้นแสง

หลักการของการสร้างภาพด้วยโปรแกรม Raytracer

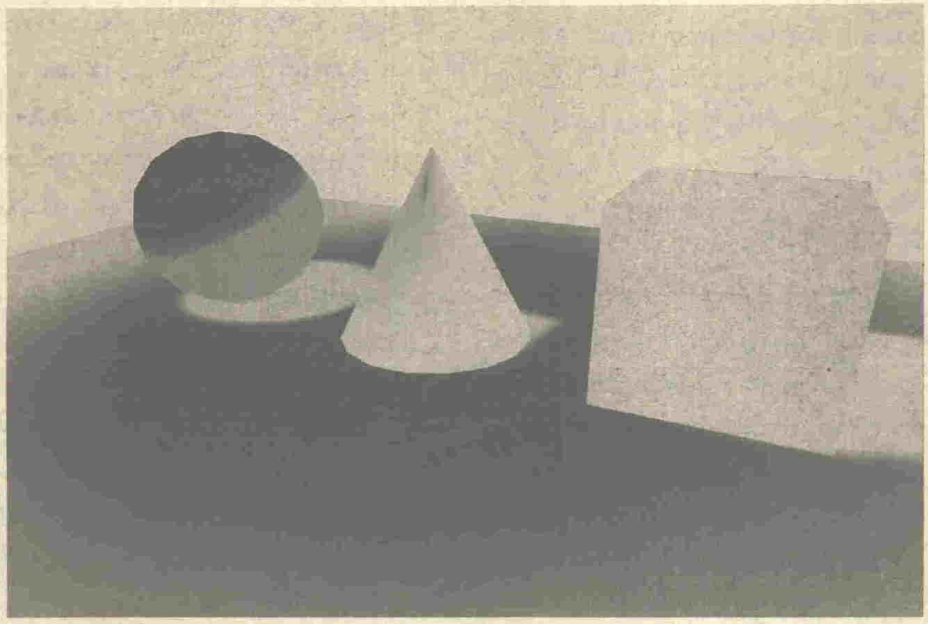
ในหลักการของการตามรอยเส้นแสงที่กล่าวมาแล้วข้างต้น เป็นการทำงานภายในของโปรแกรม Raytracer เอง ผู้ที่จะใช้โปรแกรมในขั้นต้นไม่จำเป็นต้องทราบก็ได้ ในขั้นแรกนั้นสิ่งสำคัญที่ผู้ใช้โปรแกรมต้องรู้ก็คือ การกำหนดโลกสามมิติที่จะให้โปรแกรมสร้างเป็นภาพขึ้น โปรแกรม Raytracer ทั่ว ๆ ไปจะใช้วิธีที่คล้าย ๆ กัน ซึ่งถ้าผู้อ่านเข้าใจในครั้งเดียวก็จะสามารถนำไปใช้กับโปรแกรมได้หลายตัว ขั้นตอนโดยสรุปในการที่เราจะสร้างภาพขึ้นมาสักภาพคือ เราต้องสร้าง "ฉาก" ขึ้นมาเสียก่อน ซึ่งเป็นการสร้างโลกในสามมิติขึ้นมา ต่อจากนั้นก็มากำหนดมุมมองว่าเราจะมองโลกสามมิตินั้นจากจุดใด เมื่อเราได้ฉากขึ้นมาแล้ว เราสามารถจะสร้างภาพหรือ "มอง" จากมุมใด ๆ ของฉากก็ได้

ในรูปที่ 3 จะแสดงให้เห็นโมเดลในการสร้างภาพเช่นเดียวกับรูปที่ 2 แต่จะแสดงในรูปแบบของภาพที่เหมือนจริงสักหน่อย รูปนี้ก็เป็นรูปที่สร้างจากโปรแกรม Raytracer เช่นกัน โลกสามมิติในรูปนี้ประกอบด้วยพื้นสี่เหลี่ยม มีวัตถุสามชิ้นคือ ทรงกลม กรวย และกล่องสี่เหลี่ยมวางอยู่บนพื้น มีแหล่งกำเนิดแสงอยู่ข้างบน ฉากรับภาพที่เห็นในรูปแทนจอภาพของเครื่องคอมพิวเตอร์นั่นเอง รูปที่ 4 จะเป็นภาพจริง ๆ ที่เราเห็นบนจอภาพของเครื่องคอมพิวเตอร์ ในความเป็นจริงแล้วรูปที่ 3 นั้นก็สร้างจากโลกสามมิติอันเดียวกันแต่มองในมุมที่ต่างกันกับรูปที่ 4





รูปที่ 3 โมเดลที่เหมือนจริงของการสร้างภาพโดยวิธีการตามรอยเส้นแสง



รูปที่ 4 ภาพที่เห็นจากจอภาพของเครื่องคอมพิวเตอร์ ซึ่งเกิดจากโมเดล ในรูปที่ 3

การติดต่อกับโปรแกรม Raytracer

หน้าที่ของโปรแกรม Raytracer ก็คือการสร้างภาพสองมิติจากโลกสามมิติ ดังนั้นข้อมูลที่โปรแกรมต้องการจะเป็นข้อมูลที่เกี่ยวกับโลกสามมิติที่เราจะสร้าง โปรแกรม Raytracer ส่วนใหญ่จะให้ผู้ใช้เขียนอธิบายข้อมูลต่าง ๆ ลงในไฟล์ข้อความ (Text File) จากนั้นก็กำหนดชื่อของไฟล์ให้กับโปรแกรม (เป็นอินพุตไฟล์) การเขียนอธิบายข้อมูลของโลกสามมิตินั้นมักจะมีการสร้างภาษาขึ้นมาใหม่เพื่อใช้ในการอธิบายส่วนประกอบต่าง ๆ ในโลกสามมิติโดยเฉพาะ โปรแกรมที่ผมจะแนะนำในตอนท้ายของบทความนี้จะใช้ภาษาที่คล้ายกับภาษาซี โดยจะเพิ่มส่วนของภาษาที่จะใช้กับการกำหนดโลกสามมิติขึ้นมา เช่น เพิ่มรูปแบบในการกำหนดจุดในสามมิติ เพิ่มความสามารถในการกำหนดทิศทางโดยใช้เวกเตอร์ เป็นต้น ผู้ใช้โปรแกรมสามารถที่จะใช้โปรแกรม Text Editor ตัวใดก็ได้ในการสร้างอินพุตไฟล์สำหรับโปรแกรม

การสร้างฉาก

ขั้นตอนในการสร้างฉากมีดังนี้

1. กำหนดคุณสมบัติของวัตถุที่เราจะนำไปใช้ สิ่งที่จะต้องกำหนดมีสามอย่างก็คือ รูปทรง ตำแหน่ง และพื้นผิว

2. จัดโลกสามมิติที่เราจะนำมาสร้างเป็นภาพบนจอคอมพิวเตอร์ โดยนำเอาวัตถุที่เรากำหนดไว้มาจัดลงที่ตำแหน่งที่เราต้องการ

3. กำหนดการให้แสง โดยกำหนดชนิดของแหล่งกำเนิดแสงและตำแหน่งในโลกสามมิติ

1. กำหนดวัตถุ

คำว่าวัตถุในเรื่องของ Raytracing นั้นจะประกอบด้วยส่วนต่าง ๆ สามส่วนคือ

1. รูปทรง
2. การจัดวาง
3. การกำหนดลวดลายของพื้นผิว

โปรแกรม Raytracer ส่วนมากจะมีวัตถุที่มีรูปทรงเป็นมาตรฐานอยู่แล้ว เช่น ทรงกลม กล่องสี่เหลี่ยม รูป

กรวย และทรงกระบอก โปรแกรม Raytracer บางตัวสามารถที่จะให้ผู้ใช้กำหนดวัตถุได้โดยสมการทางคณิตศาสตร์ การนำวัตถุมาใช้งานก็เพียงแต่กำหนดค่าหรือพารามิเตอร์ต่าง ๆ ที่จำเป็นเท่านั้น เช่น

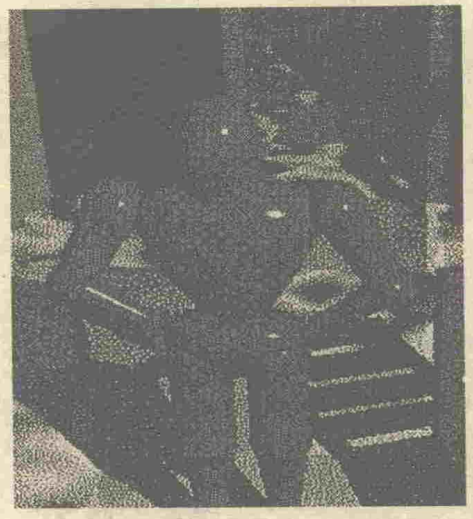
```
sphere <0,0,0,>, 2
```

เป็นการกำหนดทรงกลมขนาดสองหน่วยให้อยู่ ณ ตำแหน่ง (0,0,0)

โปรแกรม Raytracer บางโปรแกรมมีความสามารถในการรวมรูปทรงหลายรูปทรงให้เป็นกลุ่มเดียวกัน หรือสามารถนำเอารูปทรงต่าง ๆ มาสร้างเป็นรูปทรงใหม่ได้ คุณสมบัตินี้เรียกว่า Constructive Solid Gemoetry (CSG) รูปทรงใหม่สามารถกำหนดได้จากโอเปอเรเตอร์ หรือวิธีการรวมกันของรูปทรงเดิม เช่น ถ้าเราใช้โอเปอเรเตอร์ Union รูปทรงใหม่ก็จะเป็นการรวมเอารูปทรงที่นำมา Union ไว้ด้วยกัน ถ้าใช้โอเปอเรเตอร์ Intersection รูปทรงใหม่ก็จะมีเฉพาะส่วนที่รูปทรงเดิมตัดกันเท่านั้น

การกำหนดจัดวางวัตถุสามารถทำได้โดยการปรับเปลี่ยนรูปร่างจากรูปทรงมาตรฐานที่กำหนดให้มาเป็นรูปทรงที่เราต้องการ การปรับเปลี่ยนจะทำได้หลายแบบ เช่น การย้ายที่ (Translation) การขยายขนาด (Scaling) และการหมุน (Rotation) เป็นต้น

การกำหนดลวดลาย (Texture) ของพื้นผิวนั้นจะรวมถึงการกำหนดการเปลี่ยนคุณสมบัติของแสงเมื่อมาตกกระทบพื้นผิว หรือเรียกว่าการกำหนด Surface และการนำเอา Surface หลายอย่างมารวมกันเป็นลวดลาย การกำหนด Surface จะเป็นการกำหนดสี การกำหนดสี ณ จุดที่แสงสะท้อนเข้าตาโดยตรง การกำหนดการสะท้อน การหักเหของแสง ถ้าเราจะสร้างวัตถุที่เป็นกระจกเงา เราสามารถใช้รูปทรงระนาบ และกำหนด Surface ให้สามารถสะท้อนแสงได้ ส่วนการกำหนดลวดลายนั้นทำให้เราสามารถนำ Surface หลาย Surface มารวมกันได้ หรืออาจจะนำลวดลายที่มีอยู่แล้วมาทาบหรือวาดลงบนพื้นผิวได้



2. โลกสามมิติ

ในการกำหนดโลกสามมิติที่เราจะมองผ่านจอภาพออกไปนั้นค่อนข้างจะต้องใช้จินตนาการสักหน่อย และมีเรื่องของคณิตศาสตร์มาเกี่ยวข้องด้วย สำหรับผู้ที่ไม่ชินกับคณิตศาสตร์มากนักก็ไม่ต้องกังวลครับ เพราะเมื่อลองใช้โปรแกรมไปสักพักก็จะเข้าใจไปเอง สิ่งที่คนทั่วไปจะสับสนมีสองเรื่องก็คือ

- การนำเอาเรื่องการกำหนดโลกสามมิติมาปนกับการมองเห็น
- การจัดวางวัตถุในฉาก

ในปัญหาแรกนั้น วิธีแก้ก็คืออย่านำมาปนกัน เวลาสร้างโลกในสามมิตินั้นเราต้องจินตนาการว่าเรามีที่ว่าง ๆ อยู่ที่หนึ่ง ในที่ว่างนั้นมีแกน XYZ อยู่ การกำหนดโลกสามมิติก็คือการนำเอาวัตถุที่เราต้องการมาวางไว้ที่ตำแหน่งต่าง ๆ เทียบกับแกน XYZ นั้น โดยไม่ต้องกังวลว่าภาพที่เกิดจะเป็นอย่างไร เพราะหลังจากที่เราสร้างฉากเสร็จเรียบร้อยแล้ว เราจะสามารถสร้างภาพที่เกิดจากการมองฉากเดียวกันนั้นได้หลายภาพ ขึ้นกับตำแหน่งที่เรามอง ดังนั้นเมื่อคิดว่าอยาก "มองเห็น" ภาพแบบใดบนจอภาพ (สองมิติ) ก็ให้จินตนาการหรือนึกถึงว่าในโลกของความเป็นจริง (สามมิติ) ของภาพนั้นเป็นอย่างไรก่อนแล้วนำไปจัดฉาก เมื่อได้ฉากมาแล้วเรายังสามารถปรับแต่งการมองเห็นในสองมิติได้อีก จากการปรับตำแหน่งหรือมุมมอง

ในปัญหาที่สอง ต้องทำความเข้าใจว่าวัตถุที่จะใช้ได้ในฉากหรือจะพูดอีกอย่างหนึ่งก็คือ วัตถุที่โปรแกรม Raytracer เข้าใจ จะต้องมีรูปทรงที่สามารถแทนได้ด้วยสมการทางคณิตศาสตร์ เช่น ทรงกลม เป็นต้น โปรแกรม Raytracer บางโปรแกรมจะมีรูปทรงที่เป็นมาตรฐานอยู่แล้ว บางโปรแกรมยอมให้ผู้ใช้สร้างรูปทรงขึ้นมาเองได้ แต่เนื่องจากเราสามารถที่จะนำรูปทรงเดียวกันไปวางไว้ที่ตำแหน่งใดก็ได้ และจะจัดวางอย่างไรก็ได้ในฉากของเรา ดังนั้นตัวรูปทรงเองจะต้องกำหนดขึ้นให้เป็นมาตรฐาน โดยจะมีการจัดวางที่ตายตัวขึ้นกับแต่ละรูปทรง เช่น โปรแกรม Raytracer อาจจะมีทรงกระบอกอยู่ในโปรแกรมอยู่แล้ว แต่เป็นทรงกระบอกที่ตั้งขึ้นในทิศแกน Y ถ้าในภาพที่เราสร้างจะต้องมีทรงกระบอกที่วางตัวในแนวแกน X เราจะต้องทำการปรับเปลี่ยน (Transform) การวางตัวของทรงกระบอกมาตรฐานนั้นก่อน

3. การจัดแสง

เป็นการกำหนดตำแหน่งและคุณสมบัติอื่น ๆ ของแหล่งกำเนิดแสง คุณสมบัติที่เราสามารถจะกำหนดได้นั้นขึ้นกับแต่ละโปรแกรม โดยทั่วไปเราสามารถที่จะกำหนดสีของแหล่งกำเนิดแสงได้ บางโปรแกรมจะมีชนิดของแหล่งกำเนิดแสงให้เลือก เช่น แบบจุด แบบสปอทต์ไลท์ หรือแบบมีลวดลาย

การกำหนดจุดมอง

เมื่อเราสร้างฉากเสร็จเรียบร้อยแล้ว ก็จะมาถึงขั้นตอนสุดท้ายคือการกำหนดจุดมอง โดยทั่วไปสิ่งที่จะต้องกำหนดก็คือตำแหน่งของตา ทิศทางที่มอง และทิศของศีรษะ สิ่งที่ต้องกำหนดเพิ่มเติมจากนี้ขึ้นมาก็เนื่องจากเราใช้โมเดลการสร้างภาพด้วยคอมพิวเตอร์ดังที่กล่าวมาแล้ว ซึ่งจะประกอบด้วยการกำหนดความละเอียดของภาพ การกำหนดความกว้างของมุมมอง การกำหนดสัดส่วนของรูป เป็นต้น รายละเอียดในการกำหนดจะขึ้นกับแต่ละโปรแกรม ซึ่งจะยกตัวอย่างในกรณีของโปรแกรม Polyray ในตอนต่อไป

แนะนำ Polyray

โปรแกรม Polyray เป็น Raytracer บนเครื่อง PC ตัวหนึ่งที่มีความสามารถสูงพอสมควร ในแง่ของการกำหนดรูปทรง ก็จะมีรูปทรงมาตรฐานมาให้เป็นจำนวนมาก เช่น มีรูปทรงหยดน้ำกำหนดมาให้แล้ว สามารถสร้างรูปทรงใหม่โดยใช้ CSG ได้ ทั้งยังยอมให้ผู้ใช้สามารถกำหนดรูปทรงใหม่ขึ้นได้เองด้วย ในด้านของการกำหนดลวดลายของพื้นผิวก็เช่นกัน สามารถกำหนดลวดลายใหม่ ๆ ขึ้นมาแล้วนำไปใช้กับรูปทรงอื่นได้ ยิ่งไปกว่านั้นโปรแกรมตัวนี้สามารถที่จะสร้างภาพเคลื่อนไหวได้ด้วย ภาษาที่ใช้ในโปรแกรมจะเป็นภาษาที่คล้ายภาษาซี

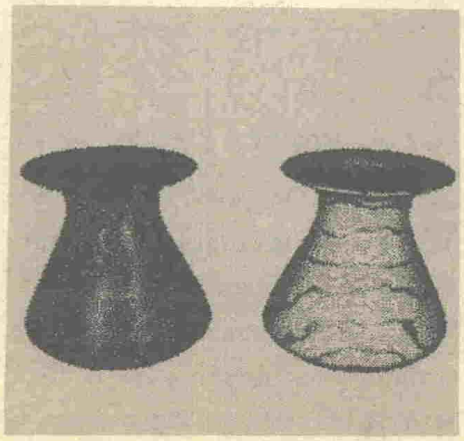
โปรแกรม Polyray จะสามารถสร้างเอาต์พุตไฟล์ได้ชนิดเดียวคือไฟล์ .TGA ซึ่งเป็นกราฟิกไฟล์แบบหนึ่ง มักจะใช้กันในโปรแกรมสร้างภาพเคลื่อนไหว จุดนี้เป็นข้อเสียของ Raytracer ตัวนี้ เนื่องจากเราต้องหาโปรแกรมอื่นมาเปลี่ยนให้เป็นไฟล์แบบที่โปรแกรมทั่วไปนิยมใช้

ฮาร์ดแวร์ที่จำเป็น

ก่อนอื่นเรามาดูกันว่าใครจะมีสิทธิ์ใช้โปรแกรมนี้ได้บ้าง ผู้ที่จะสามารถใช้โปรแกรมนี้ได้ต้องมีคุณสมบัติดังนี้ เป็นเจ้าของเครื่อง 386 DX ขึ้นไป มีหน่วยความจำไม่น้อยกว่า 2 เมกะไบต์ และต้องมีการแสดงผลเป็น VGA เป็นอย่างต่ำ

ผู้ที่มีเครื่องที่มีคุณสมบัติไม่ถึงตามเกณฑ์ก็ไม่ต้องเสียใจนะครับ ผู้เขียนเองมีเครื่องที่ตรงตามเกณฑ์พอดี ก็ยังไม่พอใจเท่าที่ควร เพราะดังที่ได้กล่าวแล้วว่าโปรแกรมประเภทนี้มีการคำนวณสูงมาก ภาพที่ซับซ้อนบางภาพ ทิ้งให้โปรแกรมทำงานทั้งคืนบนเครื่องผมเองยังไม่เสร็จเลย ผมเลยไปยืมเครื่องที่มี CPU เร็ว ๆ และมีหน่วยความจำมาก ๆ ใช้แทน

มีข้อสังเกตว่าถ้าสั่งรันโปรแกรมในดอสแล้วโปรแกรมไม่ทำงาน แสดงว่าหน่วยความจำไม่พอ หรือโปรแกรมไม่สามารถอ้างหน่วยความจำได้เต็มที่จากดอส ให้เข้าในวินโดวส์ก่อนแล้วรันจากดอสพรอมต์ของวินโดวส์



จะหาโปรแกรมได้ที่ไหน?

1. ซื้อหนังสือ Trick of The Graphics Gurus ของสำนักพิมพ์ SAM ซึ่งจะมีดิสค์แถมมาให้ ในดิสค์จะมีโปรแกรม Polyray และโปรแกรมทางกราฟิกอื่น ๆ ด้วย หนังสือราคาประมาณ 800-900 บาท

2. ถ้าผู้อ่านสามารถใช้โปรแกรม ftp ผ่านทางอินเตอร์เน็ตได้ ให้ไปที่เครื่อง

```
plaza.aarnet.edu.au
```

ในไดเรคทอรี

```
/graphics/graphics/ray/pov-dkb-archive/archive/utils
```

หรือเครื่อง

```
bode.ee.ualberta.ca
```

ในไดเรคทอรี

```
/pub/wuarchive/graphics/graphics/mirrors/ftp.informatik.uni-oldenburg.de/pub/po/utils
```

ถ้าผู้อ่านสนใจเรื่องโปรแกรม Raytracer อาจจะหาโปรแกรมอื่นมาลองใช้ดูได้จากอินเตอร์เน็ต บางเครื่องที่ผมเคยลองเข้าไปค้นเห็นมีโปรแกรม

ประเภทนี้ประมาณ 10 โปรแกรมรวมอยู่ในที่เดียวกัน

3. จากซอฟต์เมล์

ไฟล์ที่มีประโยชน์

หลังจากได้โปรแกรมมาแล้ว ให้ลองอ่านไฟล์เหล่านี้

quickref.txt เป็นสรุปรูปแบบคำสั่ง

polyray.doc เป็นคู่มือการใช้งาน

color.doc เป็นลวดลายที่ใช้บ่อย

เอกสารที่ให้มากับโปรแกรมนี้ค่อนข้างเป็นเรื่องทางเทคนิคมาก เพราะว่าโปรแกรมมีขีดความสามารถค่อนข้างมาก การจะใช้ขีดความสามารถนั้นต้องอาศัยความรู้ความเข้าใจพอสมควร อย่างไรก็ตาม ตัวอย่างที่ให้มากับโปรแกรมมีค่อนข้างมาก ดังนั้นผู้อ่านสามารถที่จะทำการทดลองกับตัวอย่างเหล่านี้ได้ จะทำให้เข้าใจการใช้งานโปรแกรมมากขึ้น

โปรแกรมอื่นที่จำเป็น

โปรแกรม Polyray จะทำหน้าที่ของ Raytracer เท่านั้น การจะนำโปรแกรมมาสร้างภาพจำเป็นจะต้องใช้โปรแกรมอื่นมาช่วยคือ

1. โปรแกรม Text Editor ที่จะนำมาใช้สร้างอินพุตไฟล์ ผู้อ่านจะเลือกใช้โปรแกรมตัวไหนก็ได้ที่ถนัด

2. โปรแกรมที่ใช้แสดงภาพบนหน้าจอ เช่น โปรแกรม vpic หรือโปรแกรมบนวินโดวส์ที่ชื่อ pman และ WinJPEG เราจำเป็นต้องใช้โปรแกรมเหล่านี้ในการแสดงผลไฟล์ .TGA ซึ่งเป็นเอาต์พุตของโปรแกรม Polyray โปรแกรมบางโปรแกรมจะมีความสามารถในการเปลี่ยนชนิดของไฟล์ให้เป็นไฟล์แบบอื่นได้ เช่น เปลี่ยนจากไฟล์ .TGA เป็นไฟล์ .GIF และ .BMP

3. ถ้าต้องการให้โปรแกรมสร้างภาพเคลื่อนไหวก็จำเป็นจะต้องใช้โปรแกรมที่สามารถแสดงภาพเคลื่อนไหวได้

โปรแกรมเกี่ยวกับกราฟิกที่กล่าวมาแล้วข้างต้นสามารถจะหาได้ง่ายจากอินเตอร์เน็ต หรืออาจจะดาวน์โหลดจาก BBS ในเมืองเราก็น่าจะมี

ตัวอย่างง่าย ๆ

เมื่อได้โปรแกรมมาแล้วมาลองใช้โปรแกรมด้วยตัวอย่างง่าย ๆ คือการสร้างภาพที่ประกอบไปด้วยวัตถุสองชิ้นคือ ทรงกลมสีแดงและแหล่งกำเนิดแสง ในส่วนของทรงกลมเราจะสร้างให้พื้นผิวมีลักษณะเป็นมันสามารถสะท้อนแสงได้ เราจะต้องสร้างเท็กซ์ไฟล์ชื่อ sphere.pi__และใส่ข้อมูลต่าง ๆ โดยเริ่มจากการกำหนดลักษณะของพื้นผิวของทรงกลม เราจะสร้างลวดลายขึ้นมาใหม่โดยจะให้ชื่อว่า shiny__red ดังนี้

```
1 define shiny__red texture {
2     surface {
3         ambient red, 0.2
4         diffuse red, 0.4
5         specular white, 0.8
6         microfacet Reitz 10
7         }
8     }
```

หมายเลขด้านหน้าของบรรทัดจะใช้ประกอบการอธิบายเท่านั้น ไม่จำเป็นต้องใส่ไว้ในไฟล์

การสร้างลวดลายแบบนี้จะต่างจากวิธีการของ Raytracer ทั่วไปที่ผู้ใช้สามารถใช้ชื่อที่โปรแกรมกำหนดมาให้เลย แต่วิธีนี้จะมีความยืดหยุ่นสูงมากกว่า เราสามารถจะสร้างลวดลายขึ้นมาเองได้ตามใจชอบ ความหมายในบรรทัดต่าง ๆ มีดังนี้

บรรทัดที่ 1 เป็นการบอก Polyray ว่าเราจะกำหนดชนิดของลวดลาย (Texture) ขึ้นมาใหม่ชื่อ shiny__red

บรรทัดที่ 2 เป็นการเริ่มต้นกำหนดชนิดของพื้นผิวในลวดลาย

บรรทัดที่ 3 จะเป็นการกำหนดว่าเมื่อไม่มีแสงจากแหล่งกำเนิดแสงในภาพมาตกกระทบจุดบนทรงกลม ให้คิดเสมือนว่ามีแสงจากแหล่งกำเนิดแสงอื่นรอบ ๆ ด้าน (Ambient Light) และพื้นผิวจะสะท้อนแสงสีแดงออกไปด้วยค่าความเข้ม 0.2 (ค่าความเข้มของทุกสีมีค่าตั้งแต่ 0 ถึง 1) คำว่า red เป็นค่าที่กำหนดมาก่อนโดยโปรแกรม Polyray ค่าอื่น ๆ ที่เราสามารถนำมาใช้ได้จะมีรายละเอียดอยู่ในไฟล์ color.doc

บรรทัดที่ 4 จะเป็นการกำหนดว่าเมื่อมีแสงจากแหล่งกำเนิดแสงมากระทบจะสะท้อนแสงสีแดงออกไปด้วยความเข้ม 0.4

บรรทัดที่ 5 จะเป็นการกำหนดสี ณ จุดที่แสงสะท้อนจากแหล่งกำเนิดแสงมาเข้าตาผู้สังเกตพอดี ให้เป็นสีขาว ตรงนี้จะทำให้วัตถุมีลักษณะเป็นมัน จุดนี้เราเรียกว่าเป็นจุด Highlight

บรรทัดที่ 6 เป็นการกำหนดให้ความเข้มของสีขาวที่จุด Highlight มีความเข้มลดลงครึ่งหนึ่งเมื่อมุมตกกระทบของแสงเป็น 10 องศา

ในการลวดลายที่มีสีด้านหรือที่เรียกว่า Matt นั้นเราจะใช้เฉพาะบรรทัดที่ 3 และ 4 เท่านั้น เมื่อไม่มีการกำหนดค่า specular ก็จะไม่เห็นการ

สะท้อนจากแหล่งกำเนิดแสง

การกำหนดลวดลายแบบอื่น ๆ เช่น แบบโปร่งใส หรือแบบกระจก สามารถจะกำหนดได้โดยการกำหนดค่าเหล่านี้ ตัวอย่างการกำหนดลวดลายแบบต่าง ๆ สามารถดูได้จากไฟล์ colors.inc ซึ่งไฟล์นี้เราสามารถ include เข้าไปใช้ในไฟล์อื่นได้โดยใช้คำสั่ง

```
include "colors.inc"
```

ต่อไปเราจะกำหนดวัตถุในภาพให้เป็นทรงกลมมีรัศมี 2 และมีลวดลายเป็นแบบ shiny_red

```
object {
sphere <0,0,0>, 2
      shiny_red
}
```

Polyray จะทราบเองว่า shiny_red คือลวดลายและสามารถนำมาใช้ในการกำหนดลวดลายของวัตถุได้

ขั้นตอนต่อไปเป็นการกำหนดแสง เราสามารถกำหนดแสงในฉากหลังได้ดังนี้

```
background skyblue
```

คำว่า background เป็นการบอกการกำหนดสีฉากหลัง

คำว่า skyblue เป็นค่าสี

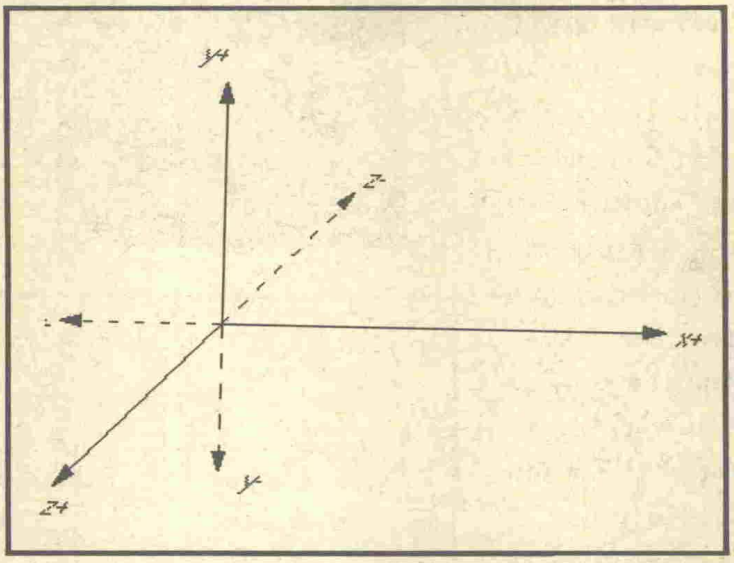
การกำหนดแหล่งกำเนิดแสงที่เป็นจุดมีรูปแบบเป็น

```
light (color,) <position>
```

ตรงส่วนของ color เราจะกำหนดหรือไม่ก็ได้ ถ้าไม่กำหนดโปรแกรมจะถือว่าเป็นสีขาว ส่วนตำแหน่งต้องกำหนดเป็นโคออร์ดิเนต เช่น

```
lignt <-10,3,-20>
```

เป็นการกำหนดแหล่งกำเนิดแสงสีขาวอยู่ที่ตำแหน่ง X = −10, Y = 3, Z = −20 ในโลกสามมิติ ทิศทางของแกน XYZ ที่ใช้ในโปรแกรม Polyray จะเป็นดังรูปที่ 5



รูปที่ 5
แกน XYZ ที่ใช้ในโปรแกรม Polyray

ขั้นตอนสุดท้ายเป็นการกำหนดจุดมอง

```
1  viewpoint {
2        from <0,0,-8>
3        at <0,0,0>
4        up <0,1,0>
5        angle 45
6        resolution 160,160
7  }
```

บรรทัดที่ 2 เป็นการกำหนดตำแหน่งตา คือกำหนดว่ามองจากจุดไหน

บรรทัดที่ 3 เป็นการกำหนดตำแหน่งที่ตามองออกไป

บรรทัดที่ 4 เป็นการบอกทิศศีรษะของผู้มอง ในที่นี้กำหนดเป็นทิศตั้งขึ้นตามแนวแกน Y เราสามารถจะกำหนดให้เอียงคอมองก็ได้ โดยเปลี่ยนค่า up ให้เป็นทิศอื่น

บรรทัดที่ 5 เป็นการกำหนดความกว้างของมุมมองจากขอบซ้ายไปยังขอบขวาของฉากรับภาพ

บรรทัดที่ 6 เป็นการกำหนดขนาดของภาพจริงที่จะมาปรากฏบนหน้าจอ ในที่นี้ภาพที่ได้จะมีขนาด 160×160 พิกเซล

หลังจากนี้ให้รวมโค้ดทั้งหมดเข้าด้วยกันในชื่อ sphere1.pi (pi ย่อมาจาก polyray_input ตัวอย่างโปรแกรมทั้งหมดของ polyray จะมีนามสกุลเป็น .pi) แล้วทำการเรียกโปรแกรม polyray ดังนี้

```
polyray sphere1.pi
```

polyray จะทำการคำนวณและแสดงสถานะการทำงานทางจอภาพ เราสามารถกำหนดให้ polyray แสดงภาพที่จะเกิดขึ้นแบบหยาบ ๆ ได้ แต่การทำงานจะช้าลง ผลลัพธ์ที่ได้จากโปรแกรมจะเป็นไฟล์ชื่อ OUT.TGA เนื่องจากเราไม่ได้กำหนดชื่อไฟล์ผลลัพธ์ลงไป การกำหนดตัวเลือกต่าง ๆ นี้จะกล่าวถึงในเรื่องของพารามิเตอร์ที่ใช้ในการรันโปรแกรมในตอนต่อไป

เราสามารถใช้โปรแกรมหลายโปรแกรมมาเรียกดูเอาต์พุตได้ ในแผ่นซอฟต์เมล์จะมีโปรแกรมชื่อ SVGA

ซึ่งรันบนดอส และโปรแกรม WINJPEG ที่รันบนวินโดวส์ โปรแกรม WINJPEG นั้นสามารถไป unzip ที่ไดเรคทอรีใดก็ได้ แล้วใช้ Windows Setup ในการเซตอัพแอพพลิเคชัน โปรแกรม WINJPEG จะสามารถอ่านไฟล์ .TGA และสามารถเก็บในรูปแบบอื่นได้ เช่น ในรูปของ .bmp ได้

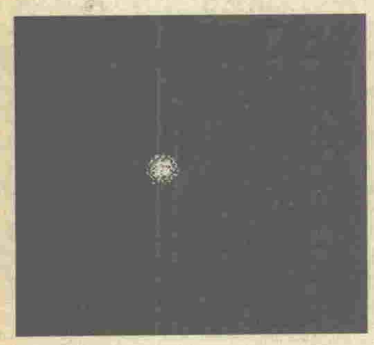
การใช้งานโปรแกรมทั้งสองนี้ขอให้ดูจากไฟล์คู่มือที่มีมากับโปรแกรม หรือจะดูจาก Help ก็ได้ รูปที่ 6 เป็นภาพที่เกิดจากอินพุตไฟล์ข้างบน

ถ้าเราปรับตำแหน่งของหลอดไฟสักเล็กน้อยแล้วเพิ่มหลอดไฟเข้าไปอีกหนึ่งดวงดังนี้

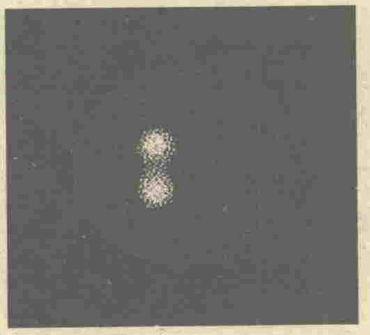
```
light <-10, 10, -20>
light <-10, -10, -20>
```

ผลที่ได้จะเป็นดังรูปที่ 7

ในรูปต่อไปผมได้นำเอาลวดลายที่กำหนดอยู่ในไฟล์ colors.inc มาใช้โดยใช้ประโยค include และใช้ลวดลายชื่อ mirror กับทรงกลม ซึ่งจะทำให้ทรงกลมมีลักษณะพื้นผิวเป็นกระจก หลังจากนั้นผมได้ใช้ความสามารถในเรื่องของ CSG ทำวงแหวนขึ้นมา โดยเอาทรง



รูปที่ 6 เอาต์พุตของ sphere.pi



รูปที่ 7
เอาต์พุตเมื่อเพิ่มหลอดไฟเข้าไปในฉาก

ลิสติงที่ 1
โปรแกรม CSG.PI

```
include "colors.inc"                    // Use colors in the file colors.inc

background <0.5, 0.5, 1>                // light blue background
light <-10,10, -20>                     // 2 lights
light <-10, -10, -20>

object {                                // mirror ball
  sphere <1, 0, 2>, 2                   // move backward
  mirror                                // from colors.inc
 }

define cylinder_z object {              // a cylinder along the line <0,0,-5>
  cylinder <0, 0, -5>, <0, 0, 5>, 0.5   // to <0,0,5> with radius 0.5
 }
define unit_sphere object {
  sphere <0, 0, 0>, 1
 }

// Define a CSG shape by deleting a cylinder from a sphere
object {
  unit_sphere - cylinder_z
  shiny_red
  translate <2, 0, 0>                   // move to the right
  rotate <0,-10, 0>                     // rotate about Y axis
  }

// Set up the camera
viewpoint {
  from <0,0,-8>
  at <0,0,0>
  up <0,1,0>
  angle 45
  hither 1
  resolution 160, 160
  }
```

กระบอกมาลบออกจากทรงกลม สังเกตว่าการกำหนดทรงกระบอกและทรงกลมนั้นจะทำที่จุดศูนย์กลางของฉากก่อน โดยใช้ประโยค define object ซึ่งเป็นการกำหนดชื่อให้วัตถุนั่นเอง ตัววัตถุจริง ๆ ที่จะแสดงบนจอภาพจะต้องขึ้นต้นด้วยคำว่า object เท่านั้น ในภาพนี้มีวัตถุสองชิ้น คือชิ้นที่เป็นลูกบอลกระจกและวงแหวน โอเปอเรเตอร์ลบ "-" จะทำให้เกิดวงแหวนขึ้น เมื่อกำหนดวงแหวนแล้วเราสามารถเคลื่อนย้าย (translate) หรือหมุน (Rotate) วัตถุได้ รายละเอียดขอให้ดูจากอินพุตไฟล์ CSG.PI ในลิสติงที่ 1

Commandline Parameter

ในการเรียกใช้งาน เราสามารถใส่พารามิเตอร์ลงไปได้ พารามิเตอร์ที่มีประโยชน์มีดังนี้

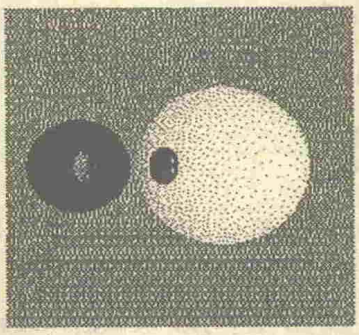
- O filename เป็นการกำหนดชื่อเอาต์พุตไฟล์

-r2 เป็นการกำหนดวิธีการสร้างภาพแบบโครงลวด Wireframe

โปรแกรมจะแสดงภาพคร่าว ๆ ได้รวดเร็วมาก ซึ่งมีประโยชน์มากในการจัดตำแหน่งของวัตถุก่อนเริ่มทำ Raytracing

-V0 งดการแสดงผลในขณะที่โปรแกรมสร้างภาพ โปรแกรมจะไม่แสดงรูปบนหน้าจอ ซึ่งจะทำให้โปรแกรมทำงานเร็วขึ้น แต่เราจะไม่เห็นรายละเอียดของรูปที่กำลังสร้าง

-R เราสามารถหยุดการทำงานของโปรแกรมชั่วคราวได้โดยกดคีย์ใด ๆ ในขณะที่โปรแกรมกำลังทำงาน เมื่อต้องการจะทำงานต่อให้เรียกโปรแกรมโดยใช้พารามิเตอร์นี้ โปรแกรมจะทำงานต่อได้



รูปที่ 8 เอาต์พุตของ CSG.PI

ตำแหน่งของพารามิเตอร์จะต้องอยู่หลังชื่ออินพุตไฟล์ดังนี้

```
polyray inputfile (option)
```

เช่น

```
polyray,csg.pi -o csg
```

ส่งท้าย

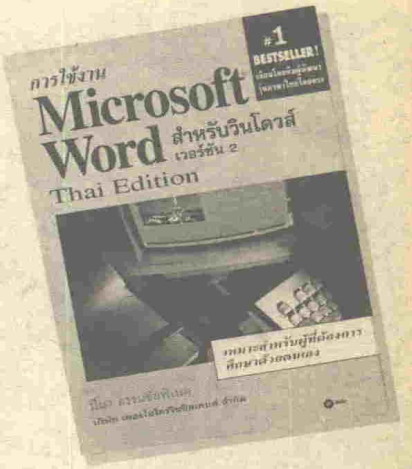
โปรแกรม Polyray มีความสามารถสูง ถ้าจะเรียนการใช้งานให้หมดก็เหมือนกับการเรียนภาษาใหม่ตัวหนึ่งทีเดียวครับ เท่าที่กล่าวมาเป็นเพียงความสามารถส่วนน้อยของโปรแกรมเท่านั้น อย่างไรก็ตาม ผู้อ่านสามารถเอาซอฟต์เมล์ไปลองเล่นหรือลองดูโปรแกรมตัวอย่างก่อนได้ ถ้าสนใจรายละเอียดเพิ่มเติมก็เขียนมาบอกกับทางไมโครฯไว้ ผมจะได้เขียนเป็นตอนต่อไป

Reference

- เมธี เผ่าบุญมี, "คอมพิวเตอร์กราฟิก", ไมโครคอมพิวเตอร์ ฉบับที่ 105, ซีเอ็ดฯ
- T. Stevens, "Fractal Programming and Raytracing with C++", M&T Books
- Sam Publishing, "On the edge of technology"
- Sam Publishing, "Tricks of The Graphics Gurus"

ตรงนี้

สำหรับผู้ที่ต้องการศึกษาด้วยตนเอง



หนังสือเล่มนี้เขียนขึ้นเพื่อตอบสนองความต้องการของผู้ใช้โปรแกรม Word ที่เพิ่มมากขึ้นตั้งแต่ Word ภาษาไทยออกสู่ท้องตลาด เนื้อหาของหนังสือเริ่มตั้งแต่พื้นฐานการใช้งานวินโดว์จนสามารถใช้งานโปรแกรมได้อย่างสมบูรณ์ เนื่องจากมีเนื้อหาครอบคลุมคำสั่งและเทคนิคการใช้ที่สำคัญ เช่น การจัดรูปแบบตัวอักษรและหน้ากระดาษ การสร้างตาราง การแบ่งคอลัมน์และการนำกราฟิกเข้ามาใช้ในการจัดเอกสาร

หาซื้อได้ตามร้านหนังสือชั้นนำทั่วไป
หรือที่ ซีเอ็ดบุ๊คเซ็นเตอร์ทุกสาขา
และสั่งซื้อได้ที่ซีเอ็ด สำนักงานใหญ่
โทร 248-6280

มีจำหน่ายแล้ว

ซีเอ็ด